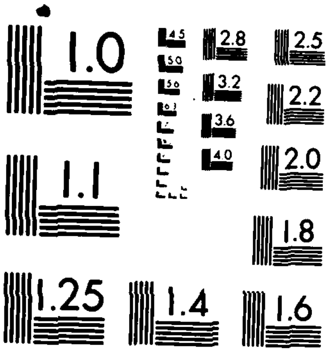
AD-A162 769 AN ANALYSIS AND SIMULATION OF THE CRAY X-MP MEMORY 1/1
SYSTEM(U) MICHIGAN UNIV ANN ARBOR DEPT OF ELECTRICAL
ENGINEERING AND CO D A CALAHAN 01 SEP 85
UNCLASSIFIED AFOSR-TR-85-1117 AFOSR-84-0096 F/G 9/2 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFOSR-TR- 85-1117	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN ANALYSIS AND SIMULATION OF THE CRAY X-MP MEMORY SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) D. A. Calahan		8. CONTRACT OR GRANT NUMBER(s) [redacted]-AFOSR 84-0096
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Michigan Dept. of Elec. Engring. & Comp. Science Ann Arbor, MI, 48109		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A3
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Bolling AFB, Washington, DC, 20332		12. REPORT DATE Sept. 1, 1985 13. NUMBER OF PAGES 7
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Presented at First Intl. Conf. on Supercomputing, Tampa, FL, Dec. 16-19, 1985.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Supercomputers,
Simulation,
Parallel processors,

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The CRAY X-MP 2- and 4-processor memory systems are analyzed and simulated using an instruction-level timing simulation of up to 16 processors. This study indicates a disturbing counter-intuitive trend to longer delays in vector accesses as both the number of processors and memory banks increase proportionately. This delay appears to be related to access start-up delays, which are determined for various memory organizations.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AFOSR-TR- 8 ~ 1117

AN ANALYSIS AND SIMULATION OF THE CRAY X-MP MEMORY SYSTEM

D. A. Calahan

Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-1109

Abstract

The CRAY X-MP 2- and 4-processor memory systems are analyzed and simulated using an instruction-level timing simulation of up to 16 processors. This study indicates a disturbing counter-intuitive trend to longer delays in vector accesses as both the number of processors and memory banks increase proportionately. This delay appears to be related to access start-up delays, which are determined for various memory organizations.

Access delays associated with running Fortran and assembly codes on an MP of up to 16 processors are studied. Projections are made which indicate that one feature of the X-MP-4 conflict resolution protocol, if used with 8 and 16 processors, creates significantly longer access delays than the X-MP-2 protocol.

I. INTRODUCTION

Research involving efficient use of commercial multiprocessor scientific architectures such as the CRAY X-MP is presently focused on algorithmic decomposition of problems into large concurrent tasks. Early evidence indicates that if the number of processors (=p) is small (say $2 < p < 16$) many problems can be decomposed into such large tasks that the speedup achieved is nearly equal to p [1][2]

At this high efficiency, a heretofore second-order effect may begin to develop importance, namely, the interference of reads and writes attempting to simultaneously access shared memory resources (memory banks, sections, etc). As p increases from the present 2 and 4 to 8, 16, and beyond, increasing the number of banks correspondingly not only increases the read/write time - for conflict checking - but also imposes severe problems in high-speed chip and memory organization, especially for memories permitting non-unit-stride vector accesses.

Recent related studies have examined this problem with real codes and a generic class of processors [3], and for the CRAY X-MP [5] with random memory fetches to gain insight into the effects of various conflict resolution protocols on access delays.

In this report, the mechanisms which account for the delay of memory accesses are studied both analytically and with the aid of an instruction-level timing simulator for the CRAY X-MP family of processors.

II. HARDWARE REVIEW

A. INTRODUCTION

Although certain definitions and observations may be appropriate to other classes of multiprocessors, this study is most directed at vector multiprocessors such as the CRAY X-MP where shared memory access rates are high for typical scientific vector codes. Indeed the motivation for this study requires some knowledge of the X-MP organization and operation. This will be reviewed below; more related discussion is given in [5] and [6].

B. X-MP-2 MEMORY AND CONFLICT RESOLUTION

Figure 1 shows the shared memory organization of the X-MP-2, the two processor X-MP extended to p processors in the simulator. For each processor, every fourth bank is accessed through the same section; with p processors, there are 4p sections. Conflicts occur at the bank or section level, as follows:

Bank-Busy conflict - The Bank Busy conflict is caused by any port within or between CPUs requesting a bank currently in a reference cycle. Resolution of this conflict occurs when the bank cycle is complete. Hold reference because of a Bank Busy conflict is 1, 2, or 3 CPs.

Simultaneous Bank conflict - The Simultaneous Bank conflict is caused by two or more ports in different CPUs requesting the same bank. Resolution of this conflict is based on a priority (see below). Hold reference is a 1 CP because of a Simultaneous Bank conflict. A Bank Busy conflict always follows a Simultaneous Bank conflict.

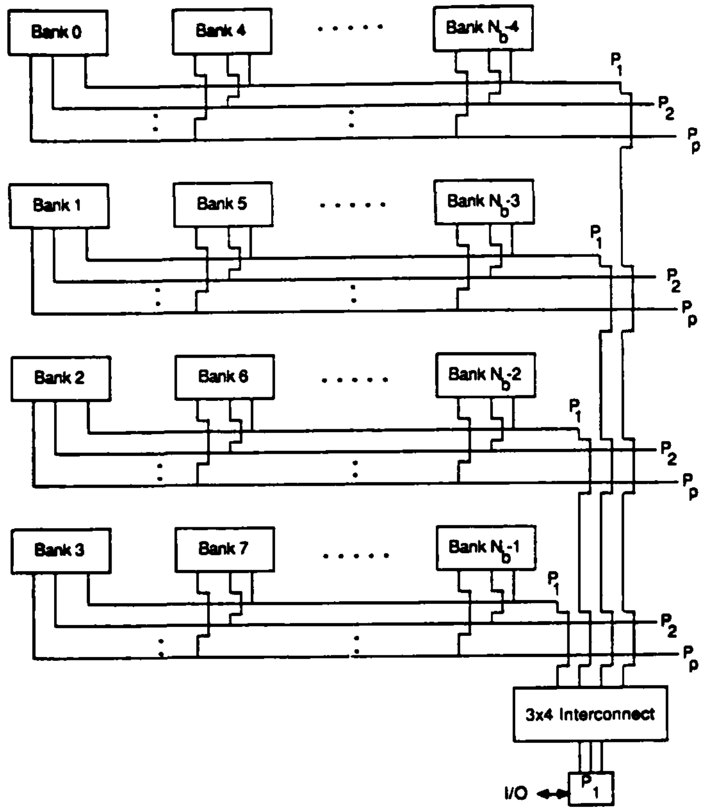


Figure 1. CRAY X-MP-2 memory organization.

Section conflict - The Section conflict is caused by two or more ports in the same CPU requesting any bank in the same section. Resolution of this conflict is based on a priority, the Bank Busy conflict, and Simultaneous Bank conflict. The highest priority port with no Bank Busy conflict and no Simultaneous Bank conflict is allowed to proceed; all other ports involved in this conflict hold (see below). Hold reference is 1 CP because of a Section conflict.

When these rules fail to resolve a conflict, the vector stride and instruction issue time are utilized to establish priority [6]. A rotating switch resolves equally-prioritized accesses between processors.

C. X-MP-4 CONFLICT RESOLUTION

The X-MP-4 (the 4-processor X-MP) conflict resolution protocol has two major differences from the X-MP-2. The effect of one of these - section numbering - is investigated in this paper. A hypothetical machine (the X-MP-2M) is defined identical to X-MP-2 except for the section numbering of the X-MP-4.

In the X-MP-2, bank #b is in section number s = mod(b,4). In the X-MP-2M, bank #b is in section number s = mod(b/4,4). This groups banks in fours (Figure 2), where each group of four belongs to one of four sections. It has been shown independently in [5] that this avoids certain catastrophic conflict patterns associated with the X-MP-2.

III. EFFECTS OF MEMORY ORGANIZATION

A. Introduction

The simulation study to be presented in Section IV has been guided by an analytical study of the effects of section numbering on performance.

This study begins by defining access delay. Let

T_F = time of attempted access of first vector element

T_L = time of access of last vector element

V_L = vector length

where times are measured in CP's. Then define the access delay

$$D_{ac} = T_F - T_L - VL + 1$$

The same definition applies both to vector reads and to writes.

This access delay cannot be measured by the X-MP hardware, but is readily obtained by simulation. Access delay produces a delay in overall program performance, but the sensitivity of this performance to access delays can vary according to the algorithm and the manner it is implemented [8]. Therefore, the access delay is the most direct measure of

```
Section #   0  1  2  3  0  1  2  3  0  1  2  3  0  1  2  3 ..
Bank #     |0  1  2  3 |4  5  6  7 |8  9 10 11|12 13 14 15|..
```

(a) X-MP-2

```
Section #       0           1           2           3           0           1
Bank #     |0  1  2  3 |4  5  6  7 |8  9 10 11|12 13 14 15|16 17 18 19|20 21 22 23|24
```

(b) X-MP-4, X-MP-2M

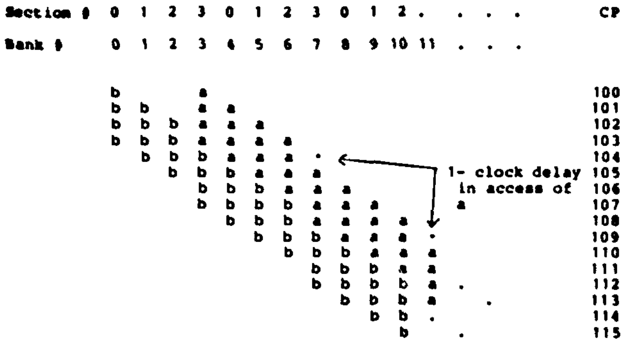
Fig. 2. Relation between section and bank numbering.

the effects of memory organization.

B. Section Conflicts

1. Introduction. In a code executing from a processor with more than one active port in the X-MP, there is a potential for conflicts between ports at the section level. For vector accesses, these occur either as (a) steady-state conflicts or (b) startup conflicts.

2. X-MP-2 steady-state section conflicts. It has been observed in [5] that in the X-MP-2, a steady-state section conflict between accesses from a CPU can occur when two ports vie for neighboring memory banks. An example is shown in the memory reservation map [10] of Figure 3 between instructions a and b. A similar example is given in [5]; it is repeated here for completeness.



Section # 0 1 2 3 0 1 2 3 0 1 2 . . . .
Bank # 0 1 2 3 4 5 6 7 8 9 10 11 . . .
CP
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
1- clock delay in access of a

Figure 3. Steady-state section conflict of neighboring accesses with X-MP-2 protocol.

At CP=100, b and a attempt to access banks 0 and 3, respectively. Because these are in different sections, both accesses are granted and banks 0 and 3 are reserved for four clock periods. At CP=101 and 102, no section or bank conflicts occur and the accesses and corresponding 4-clock bank reservations are made. However, at CP=103 b finds bank 3 reserved by a and is delayed by 1 CP. As a result, at CP=104, b and a vie for the same section. If b is assumed to have higher priority (determined by other considerations), then the next access of a will be delayed until CP=105. At CP=105, a and b are in the same relative position as at CP=100. Thus, a 1-clock delay will occur in one of every four clocks, on a steady-state basis.

Other sources of steady-state delay are given in [5].

This potentially catastrophic delay led to the recommendation in [5] that an alternate numbering of sections be adopted, and to the use of this renumbering in the X-MP-4 (see Figure 2). However, because this conflict depends on a and b accesses being from neighboring banks, it is clear that as the number of banks increases the probability of neighboring accesses from the same processor declines. The significance of this steady state conflict is correspondingly diminished. Therefore, while it may have been of concern with the 16-bank X-MP-2, it will be far less a problem in the 64-bank X-MP-4 and any many-processor extensions.

In the following sections, it will be shown the alternate section numbering system of the X-MP-4 (modeled by the X-MP-2M) has serious deficiencies as the number of processors increases and for such architectures the original numbering is superior.

3. X-MP-2M section delays

a. Introduction. For long-vector processors such as the CYBER family, vector startup conflict has traditionally been ignored. However, with a vector length of 64, the time required for a vector access to achieve a conflict-free steady state may be a significant fraction of total access time.

To investigate this phenomena, two cases are studied:

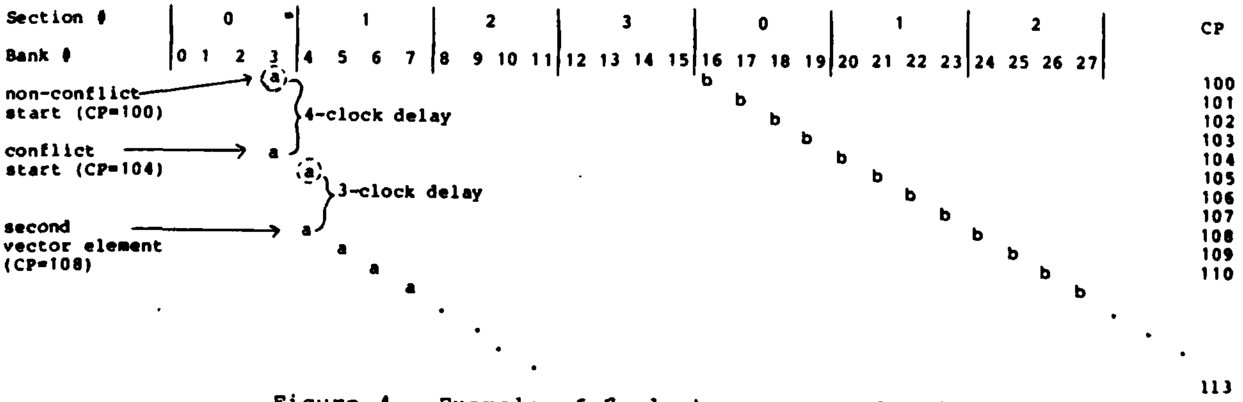


Figure 4. Example of 7-clock startup delay in uniprocesor with X-MP-2M numbering.

(1) access startup of a second port when one port is already active, and

(2) access startup of a third port when two ports are already active with conflict-free accesses.

In each case, an infinite bank memory will be assumed, so that the probability of accesses being in neighboring banks will be zero. Delays will be due solely to the periodic section numbering.

b. Two-port startup section conflicts. Figure 4 illustrates a worst case example of a two-port section conflict with X-MP-2M conflict protocol. Instruction b is in progress at CP 100 when instruction a attempts to access bank #3 (only the first clock of the access is shown). b will have priority because it is in progress, so a will hold until CP 104. Because both are in Section #0, a will then start a 4-clock access; however on CP 105 it will again conflict with b, since both are now in Section #1, causing a further three clock delay in a.

Generalizing, an instruction b in prioritized execution may be accessing bank 4r, 4r+1, 4r+2, or 4r+3 when instruction a attempts to startup in any of 16 banks in sections 0, 1, 2, or 3 (r = 4 in Figure 4). There are thus 4×16 = 64 distinctive relative startup positions of a and b. Each produces a startup delay; these may summed to produce an average startup delay. For the X-MP-2M section numbering, enumeration shows this total number of startup delay clocks is 112, yielding an average delay

$$D_{ac} = \frac{112}{64} = 1.75 \qquad (1)$$

clocks or 2.73% for VL = 64. The corresponding delay for X-MP-2 numbering is

$$D_{ac} = .25 \qquad (2)$$

c. Three-port startup section conflicts. The last section considered pairs of instructions representing a time when two ports are active. With two ports active and in the steady state, if the third port were to initiate an access, it is intuitive that, on average, it will take longer than above to find a steady state. Thus, the average startup delay can be anticipated to be longer than the 1.75 clocks of Eq. (1).

To evaluate startup conflict with three ports it is possible to set two instructions (a and b) in a conflict-free steady-state mode, and then count the delays incurred by a third instruction c initiating an access in each of 16 banks. This is repeated for all combinations of a and b in a conflict-free steady state (36 rather than the 64 of the last section). A worst case example is illustrated in Figure 5, where a 14-clock delay is indicated. Overall, among 36×16 = 576 cases, a total of 2944 delay clocks are counted, for

$$D_{ac} = \frac{2944}{576} = 5.11$$

clocks average startup delay.

d. Effects of memory design parameters. It should be noted that, unlike the steady state delay studied in section 2 above, proximity of accesses is not required, so that the same average delays will be encountered regardless of the number of banks. However, it is a function of the number of sections and the number of banks per section - both equal to four in the above study. With

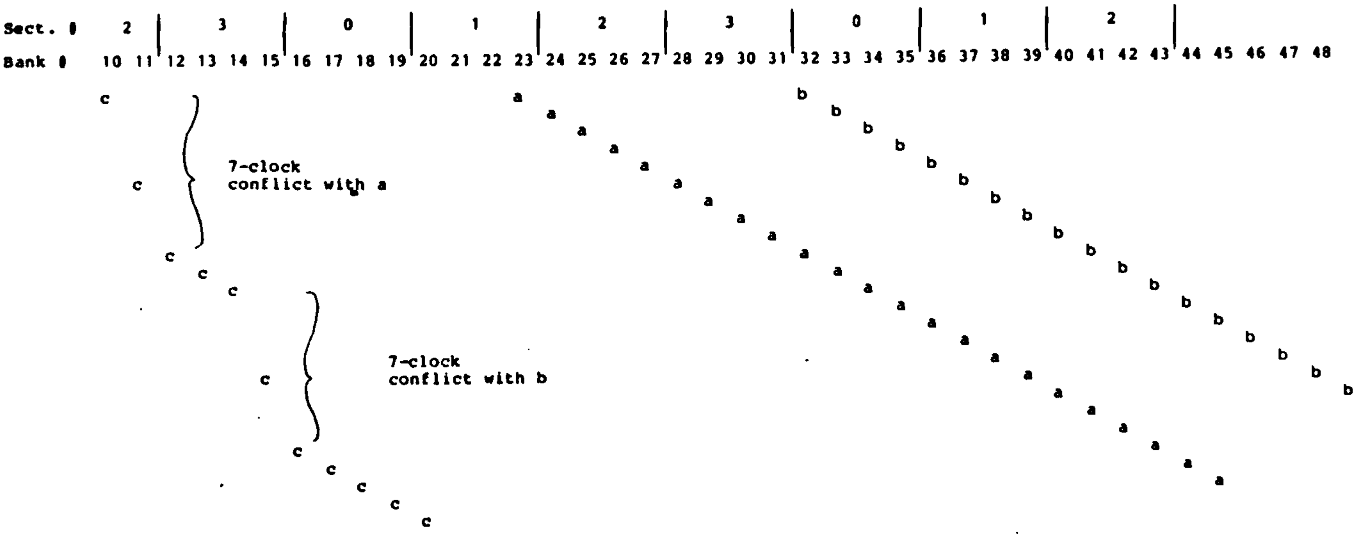


Figure 5. Worst case startup delay with three active ports; a and b in prioritized progress.

Table 1. Startup section delays as function of the number of sections (NS) and the number of banks per section (NBPS).

		Startup Delays		
NBPS	NS	2-port access (clocks)	3-port access (clocks)	
1	2	.5	---	
	4	.25	.67	(X-MP-2)
	8	.125	.28	
2	2	1.5	---	
	4	.75	4.2	
	8	.38	.86	
4	2	3.50	---	
	4	1.75	5.11	(X-MP-2M)
	8	.88	2.04	
8	2	7.50	---	
	4	3.75	11.2	
	8	1.88	4.40	
	16	.94	1.93	
16	2	15.5	---	
	4	7.75	23.5	
	8	3.87	9.11	
	16	1.93	4.15	

these as parameters, Table 1 gives the results of enumerating all combinations of instruction startups, and then averaging delays, as above. Three results are worthy of note.

(a) For a given # of banks per section (NBPS), the delay decreases as the inverse in the increase in the # of sections (NS). This occurs because conflicts are principally encountered in adjacently-numbered sections (e.g., sections #1 and #3 in Figure 5), so that adding sections (#2) merely reduces the average startup.

(b) For a given NS, the delay increases proportionately to the increase in NBPS. This is explained by the lengthening of the number of delay clocks when an access enters a wider reserved section of banks.

As a consequence of (a) and (b), if the ratio R_{bs} = NS/NBPS is maintained constant, both the two-port and three-port startups are relatively constant. If R_{bs} = 1, for example, the three-port startups of 5.11, 4.40 and 4.15 clocks are determined for NBPS=4, 8, and 16 respectively.

e. Bumped analysis. The same analysis which yielded the above D_{ac}'s can be used to evaluate the mean delay when one of two accesses, each in conflict-free steady state access, is bumped a prescribed number of clocks by an unspecified conflict. Specifically, with b in prioritized execution as in Figure 5, all 16 possible startup states of a are tested to determine which represent a conflict-free steady-state execution. For these valid states, the access of a is intentionally delayed (bumped) a prescribed number of clocks at CP100, to determine a new startup-like condition. This may result either in a being in another conflict-free access, or a may now be in a conflict with b, and an extra delay incurred. These delays are summed and averaged over all possible valid states of a and b; the results are shown in Table 2. For example, an access bumped by 4 clocks would, on the average, incur a total delay of 4 + 2.44 = 6.44 clocks before it reached a new steady state compatible with instruction b. In contrast, the X-MP-2 protocol would produce a total average delay of 4.25 clocks.

Table 2. Extra average delay suffered by a bumped access.

	Extra Delay	
Bump (clocks)	X-MP-2 protocol (clocks)	X-MP-2M protocol (clocks)
0	0.	0.
1	.25	.778
2	.25	1.44
3	.25	2.00
4	.25	2.44
5	.25	2.78
6	.25	3.00
7	.25	3.11
8	.25	3.11
9	.25	3.11
10	.25	2.33
11	.25	1.67
12	.25	1.11
13	.25	.67
14	.25	.33
15	.25	.11
16	0.	0.

IV. SIMULATION STUDIES

A. The Simulator

An instruction-level simulator produces numerical and timing information for the X-MP-2 and the X-MP-2M. The general timing accuracy of the X-MP-2 simulator is .2% for a uniprocessor and 1.3% for 2-processor hardware. Codes of conflicting only read and write instructions are exact. Five to ten minutes of CPU time are required on an Amdahl 5860 to simulate 30000 clocks in a 16-processor configuration.

B. The Experiments

Simulated bank conflict studies can be one of two forms.

(a) Accesses are initiated on a random basis across the banks, and average delays are computed using the conflict resolution protocol under investigation [5]. These studies involve no information on the relative phasing and frequency of accesses of real codes, and so yield limited insight.

(b) Complete instruction sequences derived from a real code are executed with a complete instruction-level simulator. This simulation has the disadvantages of (1) being more costly than above for the same number of accesses, and (2) incorporating accessing peculiarities of the codes (e.g., favoring certain banks). The latter may be a particular problem when, as in this experiment, the same code is executed in all processors.

The second objection was ameliorated in this study by simulating a number of codes and depicting composite average delays across all codes. Also, bank and total memory utilization were monitored and processors were initiated in a staggered manner to reduce "bunching" of high-access portions of similar codes.

Specifically, four Fortran-derived assembly codes and two directly-written assembly codes were cross-assembled and used to drive X-MP simulators, implementing X-MP-2 and X-MP-2M section conflict protocol, plus a hypothetical no-section-conflict case where all conflicts are resolved at the bank level. These codes included two Fortran matrix multiply codes, a fluids kernel and a simultaneous FFT code, in addition to an assembly-coded matrix multiply and a random read/write kernel. Nearly all vector accesses had a length of 64 and a unit stride, to reduce the number of parameters influencing delays and so to keep simulation costs reasonable.

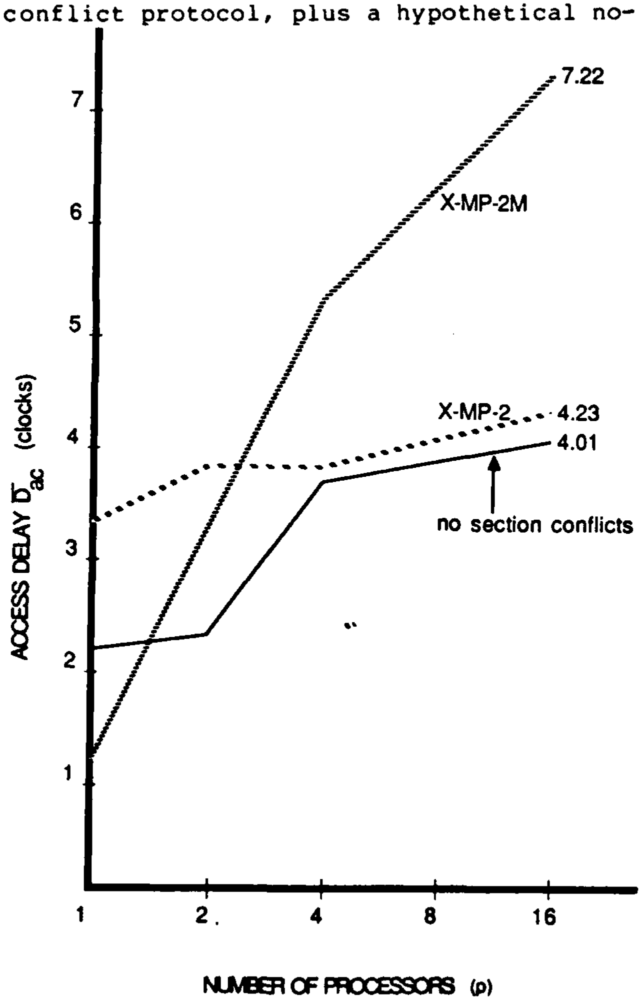


Figure 6. Simulated delays for 6-code composite.

The number of processors was varied from 1 to 16, but in the experiments to be reported here, the ratio (R_{bp}) of the number of banks to processors was maintained at 16, the common X-MP ratio for 1, 2, and 4 processors.

(C) Results

The access delay averaged across all codes ($\bar{D}_{ac}$) is shown in Figure 6. The following are worthy of note.

(1) Results for small p are most likely to include peculiarities of the accesses of a uniprocesor code rather than reflect general properties of the memory organization. Even a composite average may have unexplained behavior for small p. Thus the results for $p >$ are most insightful.

(2) There is a general trend for $\bar{D}_{ac}$ to increase with p, despite the proportionate increase in the number of banks ($R_{bp} = 16$). This applies to the no-section-conflict case as well. It is suspected that as p increases, the bank conflicts become more random, both in bank selection and phasing; such randomness is generally felt to produce the greatest number of conflicts, for a given number of accesses. An exception to this increase with p is the $\bar{D}_{ac}$ for the X-MP-2 protocol, which Figure 6 shows relatively constant for all p. This is explained by the presence of steady-state bank conflicts noted in [5] between neighboring accesses for small p. That is, steady state delays are decreasing with p while startup delays are increasing with p.

(3) X-MP-2 protocol yields a D_{ac} close to that of the no-section-conflict case for p>2, and so represents a near-optimal organization for large p.

(4) The theoretical startup delays for the two protocols (Table 1) are in ratios of 7:1 and 7.7:1, respectively, for 2-port and 3-port accesses. If the no-section-conflict plot of Figure 6 is subtracted from the other two plots, the differences -due to section conflicts - are in a similar ratio. For example, when p = 16 - the most random case - the ratio is 14.3. Being a difference ratio, this result is quite sensitive to statistical uncertainties associated with a finite simulation.

IV. CONCLUSIONS

Result (4) offers the best correlation between theoretical and simulated results in this research and gives the first strong indication that startup delays can have a dominant influence on access delays. Whether the greater influence is direct - at vector startup - or indirect via the "bumping" mechanism is unclear at this time. If the influence is direct, then only short-vector machines such as the CRAY family should be affected; however, vector restarts associated with bumping are common to long-and short-vector machines.

With an apparent relationship between startup and access delays established, work is proceeding on a mathematical formulation of the delay mechanism for large p. The principal difficulties are (1) the identification and inclusion of accessing characteristics (e.g., number and length of vectors) which contribute to total performance, and (2) the inclusion of memory organization parameters (e.g., NS and NBPS in Table 1).

ACKNOWLEDGEMENT

The author wishes to thank Ken Elliott, III, for development of the X-MP simulator. Discussions with Chris Hsiung and Al Schiffleger of Cray Research on details of X-MP conflict resolution were also helptul.

The study, exclusive of simulator development, was sponsored by Los Alamos National Laboratory and the Air Force Office of Scientific Research, under Grant AFOSR 84-0096.

REFERENCES

[1] S. Chen, J. Dongarra, and C. Hsuing, "Multiprocessing Linear Algebra Algorithms on the CRAY X-MP-2: Experiences with Small Granularity," Mathematics and Computer Science Division Technical Memorandum No. 24, Argonne National Laboratory, February, 1984.

[2] M. Moore, R. Hiromoto, and O. Lubeck, "Experiences with the Denelsor HEP," to appear in Parallel Computing, North Holland Publisher.

[3] T. S. Axelrod, P. F. Dubois, and P. Eltgroth, "A Simulator for MIMD Performance Prediction--Application of the S-1 MkIIa Multiprocessor," Report UCAL-88765, Lawrence Livermore National Laboratory, February, 1983.

[4] D. A. Calahan, "Influence of Task Granularity on Vector Multiprocessor Performance," Proc. 1984 Intl. Conf. on Parallel Processing, Bellaire, MI, August 21-24, 1984; pp 278-284.

[5] Tony Cheung, and J. E. Smith, "An Analysis of the CRAY X-MP Memory System," Proc. 1984 Intl. Conf. on Parallel Processing, Bellaire, MI, August 21-24, 1984; pp 494-505.

[6] Cray Research, Inc., "Cray X-MP Series Mainframe Reference Manual," HR-0032, Nov. 1982.

[7] P. G. Buning, and J. B. Levy, "Vectorization of Implicit Navier-Stokes Codes on the CRAY-1 Computer," Dept. of Aeronautics and Astronautics, Stanford University, November 15, 1979.

[8] D. A. Calahan, "Conflict Sensitivity of Algorithms. Part I: A CRAY X-MP Study," Report SARL #7, Dept. of Elec. Engr. and Comp. Sci., University of Michigan, March, 1985.

[9] D. A. Calahan, "Memory Conflict Simulation of a Many-Processor CRAY Architecture. Part I: A CRAY X-MP Study," Report SARL #6, Dept. of Elec. Engr. and Comp. Sci., University of Michigan, April, 1985.

[10] P. M. Kogge, "The Architecture of Pipelined Computers," McGraw-Hill, New York, 1981.